9.2 A 1-to-2GHz 4-Phase On-Chip Clock Generator with Timing-Margin Test Capability

Shunichi Kaeriyama[1], Mikihiro Kajita[2], Masayuki Mizuno[1]

[1]NEC, Sagamihara, Japan
[2]NEC, Fuchu, Japan

In recent large scale SoCs, the number of IP cores that require a dedicated clock domain is increasing, and core operating frequencies must be able to change dynamically for power reduction, etc. Although clock generators for multi-clock domain SoCs have been proposed to meet such demands [1,2], these cannot control duty ratios, which are important for high-speed clocks, and cannot generate frequencies much higher than those of input clocks because they control only rising clock edges and their output pulse widths are fixed. In response to this, we develop a clock generator in which an interleaved clock-edge-control technique makes it possible to control both rising and falling clock edges, thus making it possible to control the duty ratios of its output clocks and to extend the tuning range of output clock frequency to higher than the input clock frequency. Since this clock generator generates four-phase clocks, it can drive high-speed IP cores that require multi-phase clocks. Moreover, a clock-period dithering technique enhances the resolution of output clock frequency adjustment. In addition to this, disturbance control functions that control clock jitter, duty ratios, and clock skew among clock domains are provided (see Fig. 9.2.1). Chip testing that executes actual user programs and realistically simulates the operation of user products has become important for reducing test escapes because unexpected fault modes not covered by conventional automatic test pattern generation can be expected to increase with further technology scaling and more complex SoCs. However, it is hard to vary clock signals during such testing so as to be able to perform margin tests. The on-chip clock generator with disturbance control functions presented here makes significant timing margin testing possible.

Figure 9.2.2 is a block diagram of the clock generator circuit, which consists of an 8-phase clock generator, four 128-step phase shifters, an edge interleaver, and controller logic. The 128-step phase shifter consists of two selectors, which together perform 8-step coarse phase shifts, and a phase interpolator, which performs 16-step fine phase shifts. The phase shifter divides a clock period into 128 phases and picks out an arbitrary phase from among them in accord with a 7-bit phase code [3]. Since the phase shifter has unlimited phase shift capability, the output clock period can be shrunk or stretched by advancing or delaying the output clock phase at each clock cycle [1,2]. Conventional circuits [1-3] cannot, however, control duty ratios because they control only rising or falling edges. To test the timing margin in a chip accurately, duty ratios must be controlled properly. The combination of four phase shifters and the edge interleaver is able to do this.

Figure 9.2.3 is a timing chart for edge interleaver operations. The output of phase shifter #1 (0˚ edge) is transferred to OUT0/180 when OUT90/270 is "0". Meanwhile, phase shifter #2 changes the phase of its output (180˚ edge). This output is then transferred to OUT0/180 when OUT90/270 is "1". At this time, phase shifter #1 changes its output phase (0˚ edge). Similarly, the output of phase shifters #3 and #4 are alternately transferred to OUT90/270. The edge interleaver alternately selects only clean clock edges, and phase shifts to prepare the next clock edges are performed in the background. Thus, an unstable output state during a phase shift will not influence output clocks, and jitter generation is reduced. Edge interleaving with orthogonal clocks keeps the switching timing of the edge interleaver at the midpoint between the rising and falling edges, thus widening the tuning range of the output clock period (see the lower part of Fig. 9.2.3). Further, while the edge interleaver makes higher speed operation possible because the rate of phase shifts becomes one per clock cycle, it is still able to control both rising and falling edges. Without this interleaving, the controller logic would operate at a speed twice the clock frequency because phase codes change twice (rise and fall) with each clock cycle, and that would be impractical.

Figure 9.2.4 shows the controller logic, which provides a phase code for each phase shifter. This logic consists of three parts: jitter/dither controller logic (left side), clock period controller logic (center) and duty/phase controller logic (right side). As seen in the lower part of Fig. 9.2.4, the clock period controller logic either shrinks or stretches the clock period of the output clock while at the same time maintaining four edge-intervals of essentially equal lengths. When, for example, the timing of the 0˚ edge is advanced by ΔPeriod (i.e., the clock period shrinks by ΔPeriod), the timing of the 180˚ edge will be advanced by ΔPeriod/2. Similarly, the 90˚ and 270˚ edges will be advanced by, respectively, ΔPeriod/4 and 3ΔPeriod/4. The jitter/dither controller logic alternately changes clock periods. The duty/phase controller logic biases the timing of each clock edge. The simplicity of this logic is essential because it has to operate at a frequency higher than that of the input clock signal.

Figure 9.2.5 is an operation diagram of the clock generator. Here, as examples, 1.5625GHz, 1.25GHz, and 1.6GHz clocks are generated from a single 1.5GHz (T_{clk0} = 667ps) clock. Although the control step resolution of the clock period is limited to $T_{clk0}/128$, the average clock period is more finely adjusted by dithering the clock period (see the solid lines in the clock timing charts for Cores #1 and #2). To perform a timing margin test for Core #2, the jitter controller logic further increases the clock frequency by 5% by setting $\Delta Jitter1$ to -7 ($T_{clk2} = T_{clk0}/128\times(128+\Delta Period+\Delta Jitter1) \cong 760\text{ps} \cong 1/1.31\text{GHz}$). Since circuits driven by such further increased clock rates cannot maintain synchronization with other clock domains, clock-edge timing must be returned by stretching the clock period each time after a given number of clock cycles [4]. In this example, the clock period shrinks by 39.5ps over 4 clock cycles and then stretches by 158ps (see the dotted line in the clock timing chart for Core #2). Since the permissible clock timing range for Core #3 in sync with Core #2 will vary along with the clock timing of Core #2 (see the white band area in the clock timing chart for Core #3), the test clock frequency for Core #3 can be increased further while still maintaining synchronization. In this example, the test clock period is set to 568ps over 4 clock cycles, which corresponds to 1.76GHz (a 10% increase), and then the clock timing is returned to its previous level. Here, during 4 clock cycles out of 5, circuits are operating under severer conditions than they would be in normal operations, and the coverage of the margin test would be 80% (4/5).

Figure 9.2.6 shows measurement results for waveforms and jitter. As seen, the clock periods can be instantly changed without any distortion in the waveform. Deterministic jitter is mainly caused by nonlinearity in phase shifters. Random jitter is here roughly the same as the intrinsic jitter of the measurement system. Figure 9.2.7 summarizes the specifications and measurement results for the clock generator circuit, which is able to adjust output-clock periods, duty ratios, edge intervals between 0˚ and 90˚ clocks, and phase offsets. Typical resolution in this adjustment is 5.2ps.

Acknowledgements:
The authors thank NEC Electronics for their support throughout this work.

References:
[1] T. Fischer, F. Anderson, B. Patella and S. Naffziger, "A 90nm Variable-Frequency Clock System for a Power-Managed Itanium®-Family Processor," *ISSCC Dig. Tech. Papers*, pp. 294-295, Feb., 2005.
[2] K. Nose, A. Shibayama, H. Kodama, et al,, "Deterministic Inter-Core Synchronization with Periodically All-in-Phase Clocking for Low-Power Multi-Core SoCs," *ISSCC Dig. Tech. Papers*, pp. 296-297, Feb., 2005.
[3] S. Sidiropoulos and M. Horowitz, "A Semi-Digital DLL with Unlimited Phase Shift Capability and 0.08-400MHz Operating Range," *ISSCC Dig. Tech. Papers*, pp. 332-333, Feb., 1997.
[4] S. Fischer, R. Senthinathan, H. Rangchi and H. Yazdanmehr, "A 600MHz IA-32 Microprocessor with Enhanced Data Streaming for Graphics and Video," *ISSCC Dig. Tech. Papers*, pp. 98-99, Feb., 1999.

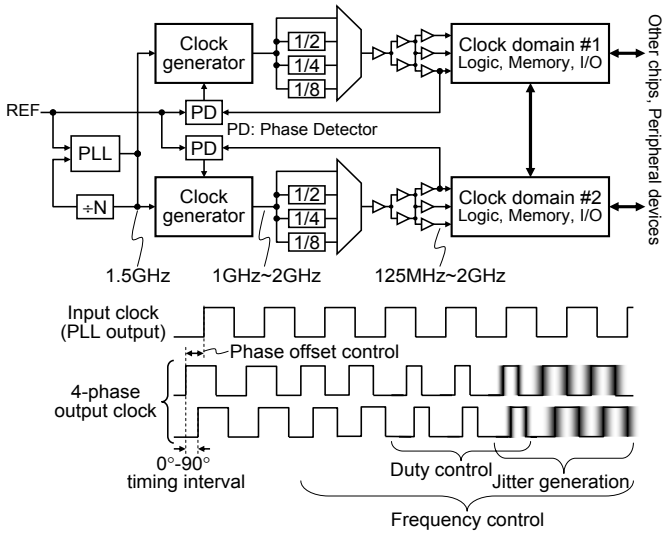


Figure 9.2.1: Clock distribution using on-chip clock generator with timing margin test capability.

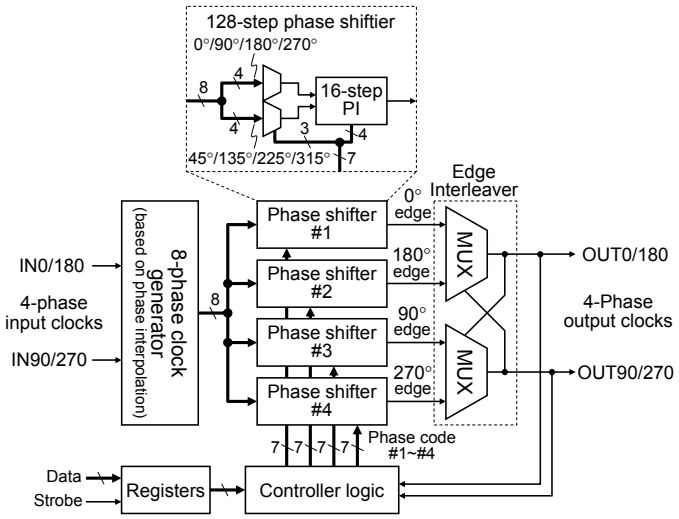


Figure 9.2.2: Clock generator circuit diagram.

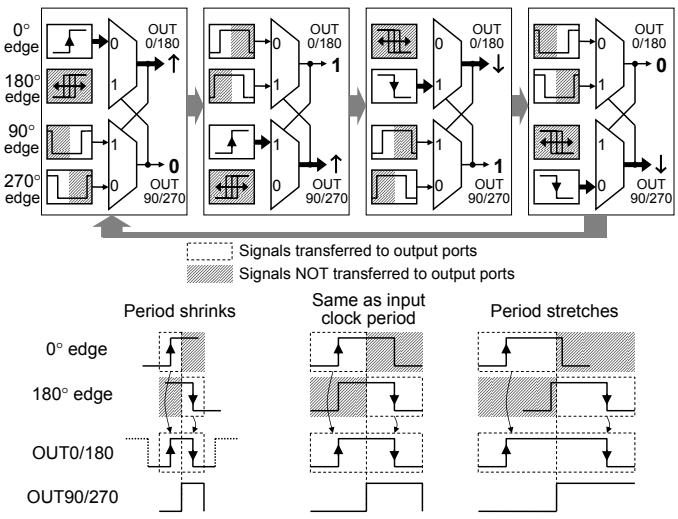


Figure 9.2.3: Edge interleaver operations.

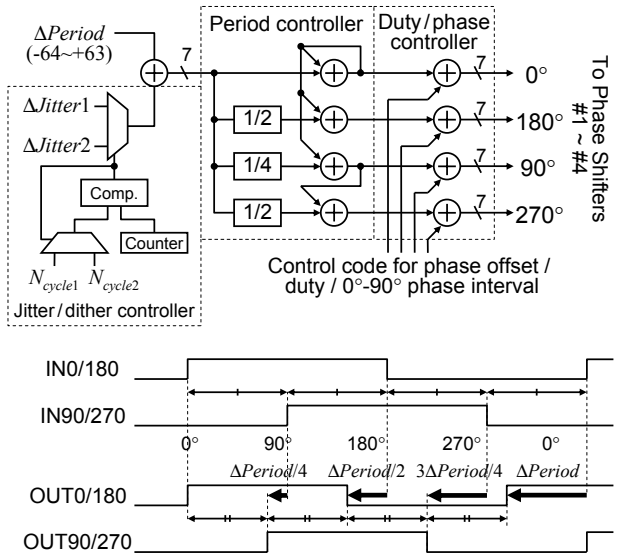


Figure 9.2.4: Controller logic: Block diagram and output clock timing chart.

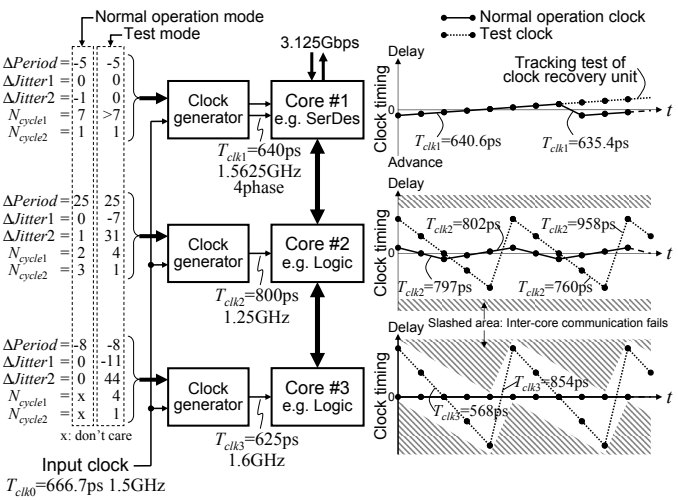


Figure 9.2.5 Clock generator operation example.

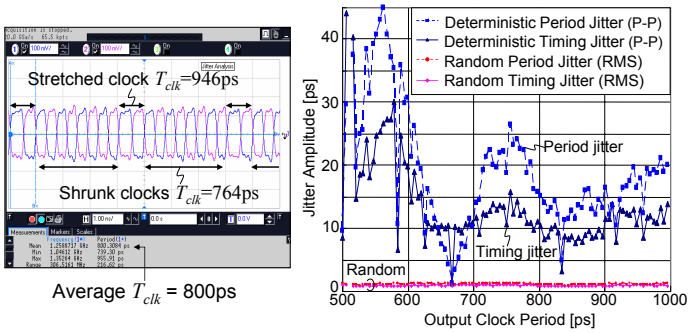
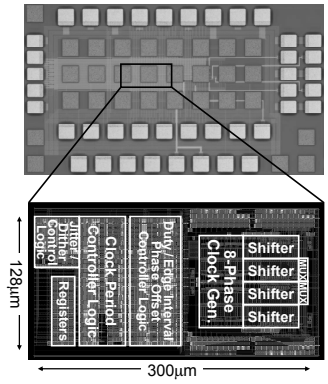


Figure 9.2.6: Measurement results.

Continued on Page 594



Process	90nm CMOS
Supply voltage	1.2 [V]
Power dissipation	40 [mW]
Occupied area	300 [μm] × 128 [μm]
Input clock	1.5GHz 4phase
Output clock	1GHz to 2GHz 4phase
Functions	•Frequency control •Jitter generation •Duty ratio control •Phase control
Timing (Phase) step resolution	5.2ps (2.8°)
Step error (5 chips measured)	±1.3ps ~ ±2.4ps (±0.7° ~ ±1.3°)
Cumulative timing error (5 chips measured)	Best chip Worst chip ±5.4ps ~ ±11.2ps (±2.9° ~ ±6.3°)
Jitter generation due to supply noise (Simulation)	Timing jitter: 60ps/V Period jitter: 110ps/V

Figure 9.2.7: Test chip summary.

